لغة (PHP)

تتميز لغة **PHP** بالكثير من الخصائص التي جعلتها الخيار الأمثل لمبرمجي الويب في العالم :

السهولة

تعتبر لغة **PHP** من أسهل لغات البرمجة تعلما، فهي تريحك من جميع تعقيدات إدارة الذاكرة وتعقيدات معالجة النصوص الموجودة في **C** من جهة، والكثير من الضعف الموجود في بينية وتصميم لغة البرمجة **Perl** من جهة أخرى.

تمتلك لغة **PHP** بنية وقواعدا ثابته وواضحة جدا، معظم قواعد اللغة مأخوذة من كل من **C** و **Java** و **Perl** لصنع لغة برمجة عالية السهولة والسلاسة دون فقدان أي من القوة في اللغة، يفيدك ذلك إذا كنت تعلم أي شيء عن لغات البرمجة الأخرى مثل **Visual Basic** أو **C** أو **Java** حيث ستجد دائما بأنك تفهم مواد الدورة بسرعة، وستكتشف كيف تقوم **PHP** بتسهيل أصعب الأمور وإذلال العقبات التي تواجه المبرمج حتى يتفرغ تماما للإبداع فقط، كل ما تفكر به تستطيع تنفيذه بلغة **PHP**.

السرعة

لغة **PHP** من اللغات المعروفة بسرعتها العالية في تنفيذ البرامج، وخاصة في الإصدارة الرابعة من المترجم، حيث تمت كتابة مترجم **PHP** من الصفر ليعطي أداءا في منتهى الروعة، كما أن لغة **PHP** مصممة أصلا كنواة لمترجم، بحيث يمكن أن تضع هذه النواة في عدة قوالب أو أغلفة لتعمل مع التقنيات المختلفة، فيمكنك تشغيل مترجم **PHP** كبرنامج **CGI** مثلا، ولكن الأفضل هو إمكانية تركيب مترجم **PHP** على مزود **IIS** في صورة وحدة إضافية تضاف

إلى المزود عن طريق دوال **ISAPI**، وتوجد نسخة أخرى منه تركب على مزود **Apache** أيضا في صورة وحدة خارجية، وتوجد أيضا نسخة مخصصة للدمج مع شفرة مزود **Apache** بحيث تصبح جزءا من برنامج **Apache** نفسه، وهي الطريقة الأكثر استخداما الآن في مزودات الويب التي تعمل على أنظمة **UNIX** وهي الطريقة التي تعطي أفضل أداء لمترجم **PHP**، حيث يصبح المترجم جزءا من المزود، وبالتالي فإنه سيكون محملا في الذاكرة بانتظار صفحات **PHP** ليقوم بترجمتها وعرضها للزوار مباشرة دون التأخير الإضافي الذي تتطلبه برامج **Perl/CGI** مثلا حيث يجب أن يتم تشغيل مترجم **Perl** مع كل زيارة للصفحة لترجمة الصفحة، ثم يتم إغلاق المترجم، ثم استدعاءه مجددا عند الزيارة الثانية وهكذا، وهذا يشكل فارقا كبيرا في المواقع ذات الضغط العالي بالذات، ويكون استخدام **PHP** حلا أفضل بكثير.

المزايا

يأتي مترجم **PHP** لوحده محملا بعدد هائل من الدوال الجاهزة الاستخدام في جميع المجالات، من دوال المعالجة الرياضية والحسابية إلى دوال الوصول إلى قواعد البيانات ومزودات **FTP**، توفر لك دوال **PHP** مثلا وصولا إلى مزودات البيانات **MySQL** و **PostgreSQL** و **MS SQL** و **Oracle** وغيرها من مزودات قواعد البيانات، وهنالك أيضا مجموعة من الدوال لمعالجة ملفات **XML**، ودوال أخرى لإرسال واستقبال الملفات عن بعد باستخدام بروتوكول **FTP**، وهنالك مجموعة من الدوال لمعالجة وإنتاج الصور ديناميكيا وملفات **Flash** ديناميكيا، ناهيك عن جميع الدوال الخاصة بمعالجة النصوص والمصفوفات.

التوافقية

كما قلنا سابقا، فعلى الرغم من أن هنالك الكثير من نسخ **PHP** التي يعمل كل منها في بيئة مختلفة، إلا أنها جميعا تشترك في النواة الأصلية التي تقوم بالمعالجة الحقيقة لملفات **PHP** لذا فإن جميع مترجمات **PHP** تتصرف بنفس الطريقة فيما يتعلق بتنفيذ السكريبتات، فإذا كان السكريبت الذي عملته يعمل على نظام **Windows** مع مزود **IIS** فيجب أن يعمل دون الحاجة لأية تغييرات عند نقله إلى مزود **Apache**، بالطبع تظل بعض الأمور البسيطة جدا التي يوفرها بعض المزودات دون غيرها، ولكن جميع البرامج التي كتبتها منذ أن بدأت تعلمي للغة إلى الآن تعمل على جميع المزودات دون الحاجة لأي تغييرات، إضافة إلى ذلك فإن التغييرات التي حدثت باللغة الأساسية من الإصدارة الثالثة إلى الرابعة قليلة جدا، وأغلب التغييرات كانت في البنية التحتية للمترجم.

الحماية

يوفر **PHP** الكثير من المزايا المتقدمة، ولكنه يوفر لك الطرق المناسبة لوضع الحدود على هذه المزايا، فيمكنك التحكم بعدد الإتصالات المسموحة بقاعدة البيانات مثلا، أو الحجم الأقصى للملفات التي يمكن إرسالها عبر المتصفح، أو السماح باستخدام بعض الميزات أو إلغاء استخدامها، كل هذا يتم عن طريق ملف إعدادات **PHP** والذي يتحكم به مدير الموقع.

قابلية التوسع

يمكنك توسعة مترجم **PHP** بسهولة وإضافة الميزات التي تريدها إليه بلغة **C**، وحيث أن الشفرة البرمجية للمترجم مفتوحة فإنك تستطيع تغيير ما تريده مباشرة لتحصل على النسخة التي تناسبك من المترجم، ويمكنك أيضا عمل الوحدات الإضافية التي تركب على المترجم لزيادة ميزاته والوظائف المبيته فيه، وفي قد قام فريق تطوير مترجم **PHP** مسبقا بعمل هذه المهمة وتحويل كمية ضخمة من المكتبات المكتوبة بلغة **C** إلى مكتبات مخصصة لتضاف إلى المترجم، ومنها حصلنا على جميع الميزات التي تحدثنا عنها مثل الوصول إلى قواعد البيانات ومعالجة ملفات **XML**.

تاريخ PHP

بدأت **PHP** كمكتبة من الدوال تضاف على لغة **Perl** لتسهل عمل برامج **CGI** بلغة **Perl**، وبعد أن تلقى **Rasmus Lerdof** بعض الاقتراحات بتحويلها إلى مترجم بسيط، قام بعمل ذلك المترجم وطرحه على الإنترنت وسماه **PHP** أو **Personal Home Pages** أي الصفحات الشخصية، فقد كان عبارة عن نسخة مصغرة من **Perl** مع بعض الميزات الإضافية للويب، ثم أضاف إليه دعما لنماذج **HTML** وسماه **PHP2/FI**، فقام مجموعة من المبرمجين بالعمل على مترجم **PHP** وأضافوا إليه واجهة تطبيقات برمجية **API** لتسهيل عملية توسعته فأصبح لدينا **PHP 3**، بعد فترة من الزمن قامت شركة **Zend** للتقنيات بعمل مترجمها الخاص للغة والذي سمي **zend** أيضا، وقد اتصف هذا المترجم بالسرعة العالية وقدراته المحسنة، وجمع مع مكتبات **PHP** الأخرى لتكوين نواة المترجم **PHP**، مترجم **PHP** الآن مقسم على قسمان: المترجم **zend** ويتم تطويره على مزودات **CVS** الموجودة في موقع **zend** والقسم الثاني يسمى **PHP** وهو عبارة عن المكتبات والدوال الأساسية التي تأتي مع البرنامج، يقوم مترجم **zend** بقراءة الملفات ومعالجتها والتعامل مع المتغيرات وتنفيذ البرنامج وتوفير واجهة تطوير للتطبيقات **API** لتوسعة اللغة، أما **PHP** فتحتوي الآن على مكتبات مكتوبة بلغة **C** ومتوافقة مع واجهة التطبيقات التي يوفرها مترجم **zend**، وبالتالي يعمل القسمان معا لتكوين مترجم **PHP**، وعندما تزور [موقع **PHP** الرسمي]() الآن وتحصل على مترجم **PHP** جاهزا أو تحصل على الشفرة البرمجية الخاصة بك، فإنك تحصل على كل من مترجم **zend** ومكتبات **PHP** معا.

تطور PHP تطورا مفاجئا في الفترة الأخيرة، وتشير إحصائيا Net Craft إلى أن مترجم PHP هو أكثر وحدات مزود Apache انتشارات على الإنترنت، كما أن مترجم PHP مركب على حوالي مليوني مزود ويب على الإنترنت.

بنية ملفات PHP

ملفات PHP هي ملفات نصية بسيطة، تشبه في تركيبها ملفات ASP وملفات HTML بشكل عام، يتكون ملف PHP من قسمان، قسم HTML وقسم PHP، الملف بالصورة الطبيعية عبارة عن ملف HTML عادي، ولكنك تستطيع تحديد أجزاء معينة من الملف ليخرج فيها الملف من وضعية HTML إلى وضعية PHP، لإخراج الملف إلى وضعية PHP توجد عدة طرق :

1 – استخدام زوج الوسوم <?php و ?> كالتالي :

```
<?php
echo 'This is PHP output!';
?>
```

2 – استخدام زوج المختصر <? و ?> وهو يستخدم بنفس الطريقة السابقة ولكنه يكون بدون الكلمة php في وسم البداية، هذا النوع من الوسوم يحتاج إلى كمية أقل من الكتابة بالطبع، ولكنه يتعارض مع وسوم xml، لذا

يقوم البعض بإغلاق ميزة الوسوم القصيرة حتى لا يحصل هذا التعارض (يمكنك اغلاق هذه الميزة بسهولة عن طريق ملف إعدادات **PHP**).

3 – استخدام زوج الوسوم **ASP**، وهو من اسمه زوج الوسوم المستخدم في ملفات **ASP** وهما <% و %>، ميزة وسوم **ASP** لا تكون فعالة بشكل قياسي ولكنك تستطيع تفعيلها عن طريق ملف إعدادات مترجم **PHP**.

4 – الطريقة الأخيرة هي استخدام زوج الوسوم التالي :

```
<script language="php" >
  echo 'This is PHP output!';
</script>
```

ولكن هذه الطريقة غير مستخدمة الآن، حيث أنها تصعب عملية التمييز بين شفرات **PHP** وباقي ملف **HTML**، وكذلك بالنسبة لبرامج كتابة ملفات **HTML** التي تعطي تلوينا للشفرة فأغلبها لا يتعرف على هذا النوع من الشفرة ويعتبره جزءا من ملف **HTML** الاعتيادي.

أفضل الطرق السابقة للتحويل إلى وضعية **PHP** هو استخدام زوج الوسوم الأول بالطبع، حيث أنه الأكثر استخدامها، ولا يحتوي على أية تعارضات كما أنه يعمل على جميع مترجمات **PHP** مهما كانت إعداداتها، ولهذا السبب سنستخدمها في جميع الأمثلة التي ستجدها في هذه الدورة.

كتابة ملفات PHP

ملفات **PHP** هي ملفات نصية بسيطة تماما كما هي ملفات **HTML**، يمكنك كتابة سكريبت **PHP** بأي برنامج كتابة نصوص يتيح لك كتابة الملفات النصية البسيطة **Plain Text** مثل **Notepad** على النظام ويندوز، ولكن أغلبية

مبرمجي **PHP** يستخدمون أدوات أخرى تسهل عليهم عملية البرمجة عن طريق تلوين الشفرات البرمجية، وتسهل عملية البحث عن الملفات واستبدال المقاطع من عدة ملفات في نفس الوقت، مثل **HomeSite**، على الرغم من أنك لن تحتاج إلى الكثير من هذه الميزات إلا أن استخدام **Notepad** في عمل ملفات **PHP** يعتبر أمرا صعبا جدا وخاصة في الملفات الضخمة حيث أن **Notepad** لا تتيح فتح الملفات الكبيرة، والمشكلة الأكبر هي أنها لا توفر ترقيما للأسطر، فإذا ظهرت لك رسالة الخطأ تشير إلى وجود خطأ في السطر **53** فلن تستطيع معرفة السطر المطلوب في **Notepad** إلا إذا قمت بالعد يدويا من السطر الأول وحتى **53** .. حسنا ماذا لو كان الخطأ في السطر **652**، يمكنك البدأ بكتابة سكريبتاتك بالبرنامج المتوفر الآن إلى أن تحصل على برنامج آخر، يمكنك بالطبع فتح ملفاتك بأي محرر نصوص، فإذا كتبتها باستخدام **Notepad** فهذا لا يعني بأنك ملزم باستخدام **Notepad** في جميع ملفاتك أو حتى في هذا الملف.

لعمل ملف **PHP** الآن قم بفتح محرر النصوص الذي اخترته وابدأ بكتابة الصفحة التي تريدها، ولا تنسى إحاطة شفرات **PHP** بالوسوم الخاصة بها، ثم احفظ الملف في أي مكان في دليل مزود الويب الخاص بك وأعطه الإمتداد المناسب **php.** أو **php3.** حسب إعدادات مزودك، ثم قم بزيارة الصفحة باستخدام المتصفح وستجد الصفحة وقد تمت ترجمتها وعرضها عليك.

تذكر بأنك يجب أن تزور الصفحة مرور بمزود الويب، ولا يمكنك عرض الصفحة عن طريق فتحها كملف خارجي، على سبيل المثال، إذا كان الدليل الجذري لصفحات مزودك هو : `C:\httpd\`

وقمت بعمل صفحة أسميتها **test.php** في ذلك الدليل، يجب أن تقوم الآن بتشغل مزود الويب وزيارة الصفحة على العنوان http://localhost/test.php، إذا قمت باستخدام الأمر **Open** من القائمة **File** في المتصفح لفتح الملف **C:\httpd\test.php** فلن ترى صفحة **PHP** مترجمة، وسترى شفرة **PHP** فقط.

تدريب

قم بتنفيذ ملف PHP التالي :

```
This is the normal html page.<br>
<?php
    echo "This is inside PHP<br>";
    echo "Hello World!<br>";
?>
```

ما الذي تشاهده عند تنفيذ البرنامج السابق؟ من المفترض أن تشاهد الخرج التالي :

```
This is the normal html page.
This is inside PHP
Hello World!
```

ها قد انتهيت من كتابة برنامجك الأول بلغة PHP، لا تقلق إذا لم تفهم أي شيء فيه، سنتعلم الآن كيفية استخدام المتغيرات والعبارات بلغة PHP.

Mimoune zakaria